DCIEM No. CR 2001-018

CAD extensions and other refinements to the LOCATE Workspace Layout Tool – Final report

by

Jack L. Edwards

Artificial Intelligence Management and Development Corporation
206 Keewatin Ave., Toronto, ON
M4P 1Z8

PWGSC Contract No. W7711-9-7600-001-TOR

on behalf of
DEPARTMENT OF NATIONAL DEFENCE

As represented by
Defence and Civil Institute of Environmental Medicine
1133 Sheppard Avenue West
Toronto, Ontario, Canada
M3M 3B9

DCIEM Scientific Authority:
Name: K. Hendy
Telephone no. (416) 635-2074

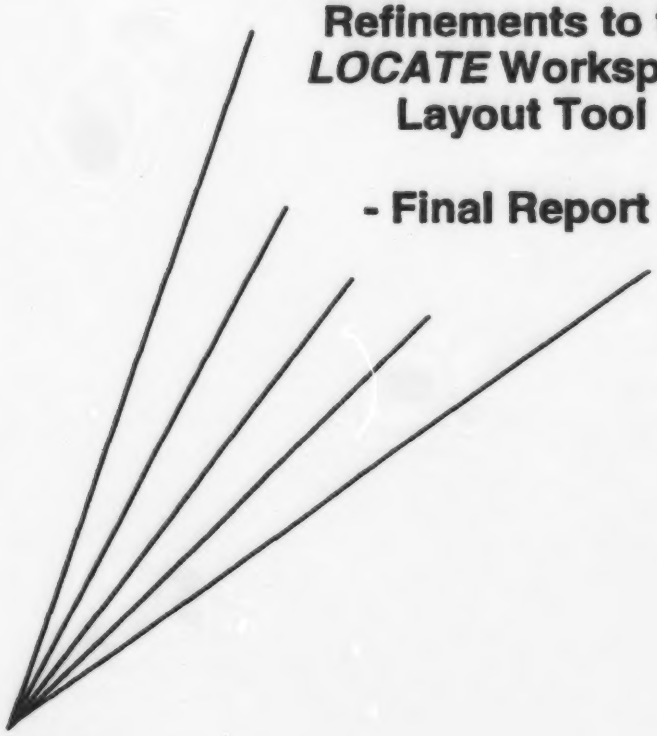
1 May 2000

Prepared for: Public Works and Government Services
Prepared by: Artificial Intelligence Management and Development Corporation
AIM (AC207, May, 2000)
Contract: W7711-9-7600/001/TOR
"CAD Extensions and Other Refinements to the *LOCATE* Workspace Layout Tool"
Scientific Authority:
Mr. Keith C. Hendy
Simulation and Modelling for Acquisition, Rehearsal and Training (SMART)
Defence and Civil Institute of Environmental Medicine

© 2000 Her Majesty the Queen in right of Canada, as represented by the Minister of National Defence

CAD Extensions and Other Interface Refinements to the *LOCATE* Workspace Layout Tool

- Final Report -



Artificial Intelligence
Management and Development Corporation
206 Keewatin Ave., Toronto, Ontario M4P 1Z8

Table of Contents

Abstract iv
Background 1
Scope of Work
CAD Extensions 3
Other Interface Refinements 3
Refinements to *LOCATE*
CAD Extensions 12
Other Interface Refinements 12
Summary Tables 13
Cost Function Summaries 15
Import and Export Commands 18
Windows
Cost Function Checks Window 21
Information Update Window 22
Links Display Window 22
More on *LOCATE* (MOL) Window 22
Obstruction (Ob) Window 22
Print Preview Window 22
Source/Receiver (S/R) Node Window 22
Tool Library Window 23
Workspace (WS) Window 23
Workstation Window 23
More Refinements
Demonstration Version and Install Script 24
PC Port 24
Alignment 24
Snap-to Grid 25
Scrolling and Nudging 25
System Checks 25
Text and the Text Tool 26

Preference Settings....26
Miscellaneous Items....27
Future Work....28
References....36

APPENDICES

Appendix A: Annotated List of *LOCATE* Source Code Files.... A-1
Appendix B: Revised Work Items for the Current Work Phase.... A-9

FIGURES

Figure 1. An Example Portion of *LOCATE's* Link Function Summary Table for the Auditory Domain....14
Figure 2. An Example Portion of *LOCATE's* Priority Weight Summary Table for the Visual Domain....15
Figure 3. Example Cost Function Output in *LOCATE's* Colour-Coded Display....16
Figure 4. Example Colour-Coded Display of Pairwise Cost Function Values....17
Figure 5. *LOCATE's* Import Options Window....19
Figure 6. *LOCATE's* Export Options Window....20

Abstract

Work reported here added a number of features to the *LOCATE* Workspace Layout Design Tool. One key feature extended *LOCATE's* ability to interface with other CAD packages through the standard DXF file format. Most notably, generic objects created in *LOCATE* now can be exported from and imported into the application.

Important interface refinements included the addition of large data summary tables that allow *LOCATE* users to enter and modify link function, obstruction function and priority weight data. Support for importing and exporting those and other attribute data was incorporated into this version of the tool. Such data can be modified in third party spreadsheets and brought back into *LOCATE*. Equally important was the combination of *LOCATE's* numerical cost function results with its colour-coded cost displays, which now provides for more efficient and effective user analysis of communication costs. Many other additions are described.

All of those additions were made in ways that continue to support complex data structures necessary to the tracking of user actions, goals and plans, which in turn serve as a basis for intelligent aiding to users of the *LOCATE* application. This is important given that *LOCATE* is the key focus of other work on an intelligence testbed, which studies intelligent aiding and agent technology.

Finally, a variety of features were identified for possible future implementation.

Résumé :

L'outil d'implantation des locaux de travail LOCATE est un outil stabilisé qui est utile dans des applications de conception pratiques. Il a été utilisé pour évaluer des configurations de passerelle pour un destroyer de classe Tribal (DDH280) des Forces canadiennes, pour analyser des conceptions de navire proposées pour la Marine américaine dans le cadre de son projet Surface Combatant-21 (SC-21), et on envisage actuellement de l'utiliser dans le cadre des travaux du Centre canadien des opérations régionales. L'outil pourrait aussi permettre de planifier l'aménagement de bureaux et d'usines de fabrication, de postes de commandement, du poste de pilotage d'aéronefs et même d'écrans d'ordinateur. Le travail visé par ce contrat comprenait des améliorations à l'outil d'implantation des locaux de travail LOCATE. Il fallait notamment améliorer la capacité de LOCATE à utiliser des fichiers créés à l'aide d'autres logiciels de CAO et à créer des fichiers de sortie semblables qui seraient ensuite utilisés dans ces autres logiciels de CAO. Un autre objectif important était d'étendre et d'améliorer certaines fonctions particulières de LOCATE. Il fallait notamment étendre les résumés généraux des fonctions de liaison et d'obstruction et des coefficients de pondération relatifs à l'ordre de priorité pour permettre aux utilisateurs d'examiner, de comparer et de modifier ces données sous une forme pratique. Enfin, on a utilisé des spécifications pour des structures de données C++, comme celles utilisées pour les objectifs, les plans et les modèles, dans le cadre d'un effort continu visant à créer une infrastructure permettant de supporter un logiciel LOCATE vraiment intelligent.

Background

The *LOCATE* workspace layout tool is a mature tool useful for practical design applications. It has been used to evaluate bridge configurations for a Tribal class (DDH280) destroyer of the Canadian Forces, to analyse proposed ship designs for the US Navy as part of their Surface Combatant—21 (SC-21) project and is currently being considered as part of work at the Canadian Regional Operations Centre. Other potential application areas include office and manufacturing plant layouts, command posts, aircraft cockpits and even computer screens.

LOCATE is both a Computer-Aided Design (CAD) environment that allows users to create workspace or facility layout designs and a programme for evaluating the communication efficiency of those designs. The term "*LOCATE*" refers, therefore, to a Graphical User Interface (GUI) and an underlying programme for analysis. *LOCATE* also supports the import and export of standard DXF files created by and for other CAD programmes such as AutoCAD.

Unlike other facility layout software, *LOCATE* allows for the analysis of any combination of four different types of communication, specifically, vision, audition, touch and movement. Designers can input information about location, orientation and the structure of elemental workstations in a workspace layout and then have *LOCATE* generate cost functions, which are measures of workspace communication efficiency. Different configurations may be tried and their cost functions compared, as a way to help users determine the best (most efficient) configuration for a design.

Originally written in DEC VAX Fortran 77 with all data entered from a keyboard, *LOCATE* was very labour intensive. Recently, it has been enhanced with a GUI, using Neuron Data's[1] Elements Environment software tools.

Intelligent aiding has been added to *LOCATE*, supported by an underlying rule and object-based expert system. An introductory slide presentation and online help are available on the World Wide Web.

LOCATE currently runs on Macintosh and PC platforms and generates two types of output files: those readable by *LOCATE* alone and those readable by both *LOCATE* and other CAD packages such as AutoCAD, as described above.

---

[1] Now Blaze Software

The purpose of the current contract was to extend *LOCATE's* interface support for the sharing of information with other CAD packages and to expand other features that would make *LOCATE* more appealing to its users. In addition, work on *LOCATE's* intelligent aiding infrastructure was extended by continuing the implementation of the complex data structures necessary to such aiding.

Scope Of The Work

Work for this contract comprised refinements to the *LOCATE* workspace layout tool. A key concern was extending *LOCATE's* ability to deal with files created in other CAD packages and with creating similar output files for use in those same packages.

A second important goal was to expand and refine selected features of *LOCATE*. One feature of particular interest was the overview summaries of link and obstruction functions and of priority weights, which would allow users to examine, compare and edit data in a convenient form.

Finally, specifications for C++ data structures, such as those for goals, plans and models, were used in a continuing effort to build the kind of infrastructure needed to support a truly intelligent *LOCATE*.

CAD Extensions

LOCATE can import and export files using the industry standard DXF file format. The application provides a user with information about a target import file's likely size, based on an analysis of its contents. It then allows the user to set a scaling factor as a way of accommodating the size of that file to preferred dimensions for a new *LOCATE* workspace. Those options are provided to the user just prior to importing the DXF file.

Recent work with import files revealed a problem with the scaling of some of *LOCATE's* generic objects, which needed adjusting, especially when the size of the imported file is much larger or much smaller than the user-specified *LOCATE* workspace file. Other needed extensions included support for exporting and importing of those generic objects.

Other Interface Refinements

Categories of items for further development are listed below. The items are identified as Essential (E) and Optional (O). Essential items were to be completed during the course of the current contract; the feasibility of incorporating Optional items was to be explored and, where practical and desirable, incorporated into *LOCATE*. The addition of all items was seen as a way to enhance *LOCATE's* appeal as an effective and desirable tool for designers.

Essential items appear before Optional items in each of the categories below and items within categories are listed, for the most part, in the order of their importance. The categories are listed alphabetically. Both Essential (E) and Optional (O) items emerged as requirements from previous developmental work.

Alignment

- Provide support for the alignment of objects in *LOCATE's* workspace; (E)
  Alignment should not permitted for objects contained within other objects, e.g., Elemental Obstructions (EObs) contained in two or more Elemental Workstations (EWs) or generic objects contained in two or more Fixed Obstructions (FObs).

Colour Coding Defaults and Committed Values

- Amber and green colours are to be used to indicate that a value is a default value (amber) vs. a user-entered or user-committed value (green). (O)
  These colours are to be used in new summary windows for Link Functions (LFs), Obstruction Functions (Obs) and Priority Weights (PWs) (see "Summary Windows", below). In the Cost Function Checks window, only "uncommitted" default data are shown.

Demo Versions of *LOCATE*

- Build a demo version of *LOCATE* for the Mac and for the PC (E).

  This was not part of the original requirements but emerged as a high priority requirement during the course of the current work.

Grid (Snap-to)

- Add a snap-to grid capability. (E)

Importing Link Function, Obstruction Function and Priority Weight data

- Support the transfer of Function and Priority Weight data among designs (E).
  - Allow users to import Link Function, Priority Weight and Obstruction Function data from other designs; both designs match in terms of the number and names of their EWs;
  - In the case where there is more than one EW, EOb or FOb, all must have unique names.

Install script

- Define requirements for creating an install script for *LOCATE* for Mac and PCs; (O)
- Implement for the Mac first, then the PC. (O)

These were optional items in the original requirements but emerged as high priority requirements during the course of the current work.

Interface Consistency

- Explore and identify rules for consistency among the interfaces of various DCIEM tools. (O)

Numerical Cost Function Results Matrix

- Add names and numbers of EWs for rows and numbers for columns; (E)
- Produce a "Total" matrix to add to the output of the existing numerical matrices; (E)
- Provide link quality and weighted costs, similar to the colour cost display; (O)
- Display by domain, as in the colour cost displays; (O)
- Delete information about location and rotation of EWs; (O)
- Explore optional display for scientific notation. (O)

Obstructions

- Explore the moving of obstructions in and out of EW's and have *LOCATE* redefine them as Elemental Obstructions (EObs) and Fixed Obstructions (FObs), as appropriate; (O)
- Model the user as an obstruction; (O)
- Make the S/R node a special item that only can be grouped manually; (O)
- Review different numbering system for EObs (e.g., 1-1; 1-2; 2-1; etc.). (O)

Palette

- Fix problem with Control–Clicking an already selected item in the tool library; (O)
- Make tool icons more consistent with items placed in *LOCATE's* workspace; (O)
- Explore the use of colour 3D icons; (O)
- Explore the use of additional (floating) palettes, tied to problem domains. (O)

PC Port

- Port *LOCATE* to the PC with all features implemented under the current contract. (O)

  This was an optional part of the original requirements but emerged as a high priority requirement during the course of the current work.

Preference Settings

- Create a "Preferences" file with selected setting options (E).

Printing

- Fix multiple refresh problem; (E)
- Clean up performance problem when printing is cancelled. (O)

Renaming Problems (EWs; EObs; FObs)

- Explore refinement of Motif's default behaviour when creating or renaming items in pop-up menus. (O)

Rotation

- Provide a rotation mode to allow for multiple rotations. (O)

Scrolling

- Dragging objects off the screen in any direction should cause automatic scrolling in that direction. (E)

Selecting and Automatic Move to Selected Objects

- Coordinate selected objects in windows with objects in the design space; (O)
- Provide the user with control over whether EObs (and other objects) expand from the top left corner or the centre. (O)

System Checks

- Update (or close) all open windows when changes are made to the active domain weights; (E)
- Check for values of domain weights other than 1's; (E)
- Check for negative domain weights; (E)
- Check for negative values for numbers; (E)
- Check for commas instead of periods in numbers. (E)

Text and Text Tool

- Fix problems associated with current text editing; (E)
- Fix problems with current copy and paste of text objects. (E)
- Allow changing of text by clicking in text box without clicking text tool first; (O)
- Allow resizing of text object using handles; (O)

Tool Library

- Delete the choice of the "Selection Tool" from the tool library. (E)

User Manual

- Extend *LOCATE's* Quick Start Manual to reflect features added in this contract. (O)

Windows

Cost Display (CD) Window

- Output of colour cost displays should include design title, domain, cost function, link quality or weighted cost view, and normalised or non-normalised view; (E)
- Add an "All" to display pop-up inside each colour CD window, supporting similar functionality to the "Close All" button; (E)
- Examine the need for a Cost Function Value Display inside each colour CD window; if appropriate, implement; (O)
- Explore how the display of percentage values might be computed for the cells in CD windows; if feasible, implement; (O)
- Explore incorporating the Display Editor into the Costs Display Windows; (O)
- Allow for the selection of a group of cells, using click-hold-drag, and for their associated names and numbers to be highlighted. (O)

Cost Function Checks (CFC) Window

- Extend earlier work on the display of default data in the Cost Function Checks Window. (E)

Elemental Workstation (EW) Window

- Update already open EW Windows when another EW is double-clicked in the workspace; (E)

- Refine the Priority Weight (1st-Order) portion of the window showing only one panel. Prompt the user that the display is for 1st-order PWs, which are entered for the Distance Domain only. (O)

Information Update (IU) Window

- Determine rules to govern the display of the IU window. (E)

  Consider the following conditions under which that window might be shown:

  1. When Smart Help is active.
  2. Once only, prior to each successful attempt to run a Cost Function (CF).

Links Display (LD) Window

- Correct minor problems with the display in this window. (E)

More On *LOCATE* (MOL) Window

- Correct a minor display problem in this window. (E)

Obstruction (Ob) Windows

- Make the x-scale and y-scale values in the Obstruction windows non-editable. (E)

S/R Node (S/R) Window

- Add windows for S/R nodes, if appropriate. (E)

Summary Windows

- Create summary windows for 1) Link Functions; 2) Obstruction Functions; and, 3) Priority Weights; (E)
- Make the windows scrolling and the values in the windows editable; (E)
- Windows should contain:
  - pop-up menus for link function selection; (E)
  - A column that is either blank or contains a "•." The bullet will indicate that there is a note associated with the decision about the function and weight in that row. In either case, the user should be able to double-click on the cell to display a window that allows the user to enter or edit a note. (O)
- Incorporate checks that PWs range from -1.00 to +1.00; (E)
- Explore incorporating "Cost Function Value Display" and "Save CF button; (E)
- Add a Commit column having a check box for committing the values in the Link Functions and Obstruction Functions windows; (O)
- Improve the efficiency when opening these windows. (O)

Workspace (WS) Window

- Close any open CD Windows whenever a user modifies the communication domains to be analysed. (E)

Workspace Boundary Tools

- review and, as appropriate, modify corners for walls, circular walls, and, positions in walls for windows and doors. (O)

Zero Point of Workspace

- By default, the zero point is set to the centre of the WS. When changing workspace dimensions, update the selected zero point; (E)

- Highlight the radio button defining the workspace zero point; (E)
- Fix problems associated with revising a workspace's dimensions to create a smaller workspace after objects have been created that will fall outside the new workspace. (E)

Miscellaneous Items

- Identify and fix intermittent problems with function graphs; (E)
- Hide Mac menus in Mac version; (E)
- Add a version number to the Get Info Window in the Macintosh. (O)

Refinements to *LOCATE*

CAD Extensions

As indicated above, *LOCATE* can import and export files using the industry standard DXF file format. Information about the target file's likely size, based on *LOCATE's* analysis of its contents, is provided to the user and that information becomes part of a decision about appropriate scaling factors for the file to be imported.

Recent work revealed a problem with scaling of some of *LOCATE's* generic objects, which required adjustment, especially when the size of the imported file is much larger or much smaller than the user-specified *LOCATE* workspace file.

That problem has now been solved by eliminating the creation of a default size for any object. That is, the user now selects the palette item for an object and clicks and drags to create an instance of the object in the workspace. Previously, if the user simply clicked in the workspace after choosing the icon, an instance of default size was created. The difficulty with that approach was that it assumed standard scale and zoom factors.

Another extension to *LOCATE's* DXF file capabilities was the ability to export generic objects, which included all items in *LOCATE's* palette such as chairs, desks, etc. as well as objects created with *LOCATE's* general drawing tools, that is, objects created through the use of lines, rectangles, polygons, etc.

Other Interface Refinements

During the course of the work on this contract, a number of items, not originally part of the contract, emerged as high priority requirements for *LOCATE*. In part, this was a result of continuing work with the US Navy, which has been employing *LOCATE* to help in the design of its warships for the 21st century and, more recently, in the exploration of the communication efficiency of designs for a command centre for amphibious support craft.

Some other items were identified which, it was believed, would add more of a commercial flavour to *LOCATE* and help facilitate its use among workspace designers and human factors experts. This was a result of British interest in the tool and a desire to have a version that could be demonstrated to interested parties in the UK.

The majority of the additions are of the latter type and include the construction of a demonstration version of *LOCATE*, an install script to ease the installation process and a port of the *LOCATE* Workspace Layout Tool to the PC. There are several items of the former type, not the least of which is the addition of an ability to export and import comma-delimited files for use with spreadsheets such as Microsoft Excel.

An annotated summary of *LOCATE's* source code files for all of refinements made as part of this contract appears in Appendix A.

Recent work with the US Navy involved the transfer of priority weight data from spreadsheets, created in a working session with Navy personnel, to *LOCATE* files for analysis. The approximate six hours of effort required for that transfer have now been reduced to a few minutes using *LOCATE's* new import command for comma-delimited files.

A complete list of the new items added to this contract, those deleted and the original items that remained is presented in Appendix B. The interface items that constitute that final list are now discussed.

Summary Tables. Of all the interface refinements in this study, this is likely the most useful. Whereas in previous versions of *LOCATE*, data were entered for individual workstations and obstructions, they may now be observed, entered and modified in large summary tables. Those tables have been implemented for both link function and priority weight data for the following *LOCATE* objects:

- Workstations;
- Elemental Obstructions
- Fixed Obstructions

and for the four communication domains and combined totals for all domains supported by *LOCATE*, as follows:

- All Domains;
- Auditory;
- Distance;
- Tactile;
- Visual

Figure 1 shows an example of a portion of a summary table for link functions for all workstations for the auditory domain. The names and numbers of workstations appear in the

first column. The second column identifies the source and receiver aspect of the link function and the third column the distance and angular component for each receiver and source aspect.

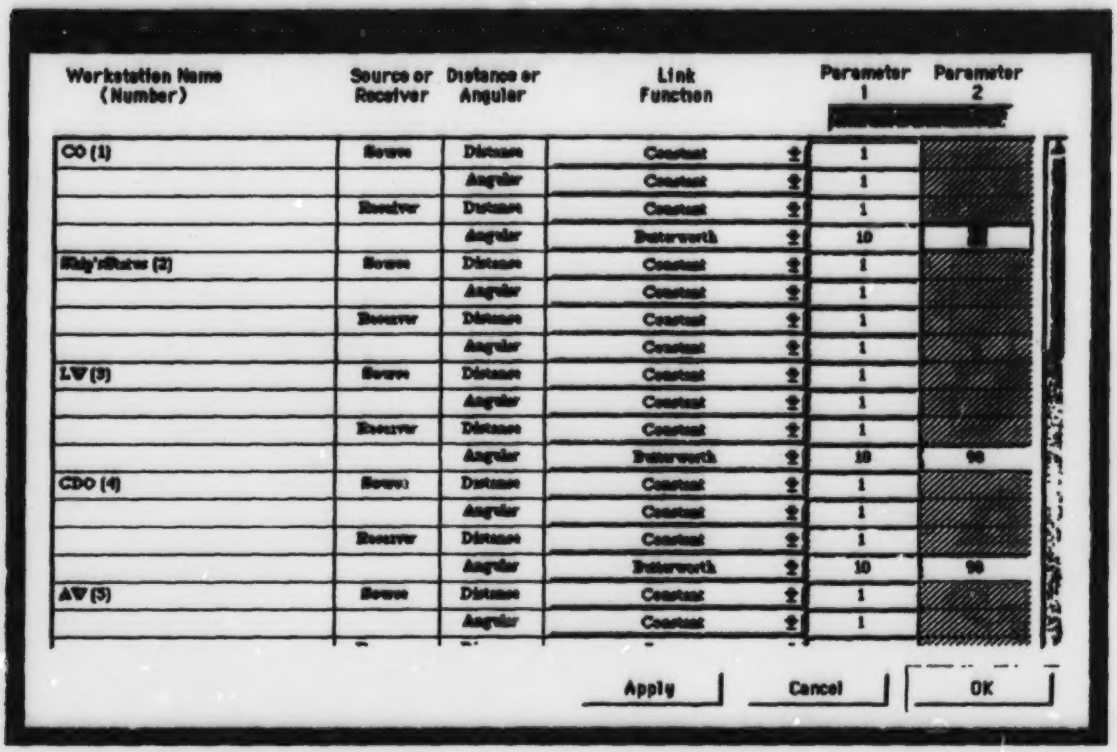


Figure 1. An Example Portion of *LOCATE's* Link Function Summary Table for an Auditory Domain

Column four shows the types of function chosen to characterise, in this example, auditory communication for the aspect and component of each workstation and column six and seven show argument values for the functions chosen. Since arguments are different, depending on the function, the names of the arguments change depending on where the cursor is located and appear in the space at the top of the window under "Parameter 1" and "Parameter 2."

Figure 2 illustrates a summary table for priority weight data for the visual domain. Workstations, as receivers of information, appear in the rows and are repeated, as sources of information, in the columns. Although the matrix may be symmetrical, most often it is not, as with, say, command structures, which by their nature imply asymmetry.

The values in the table reflect the range of priority weights, falling between -1.00 and 1.00. As Figure 2 illustrates, *LOCATE* allows a user to hide zero values, which can reduce clutter in the display and facilitate interpretation.

Source / Receiver	CO 1	Ship's... 2	LV 3	CDO 4	AV 5	SpecE... 6	SSM2 7	SSM1 8	SV 9	SpecT... 10	ExtPlat 11	Com. 12
CO - 1				1		0.5						
Ship'sStatus - 2	0.7		0.8	0.7	0.8	0.7	1	1	0.8			
LV - 3	0.8			0.8	0.7	1			0.7			
CDO - 4	1					0.5						
AV - 5	0.9		0.7	0.9		1			0.7			
SpecEval - 6	0.5		1	0.5	1				1			
SSM2 - 7	0.6		0.9	0.6	0.9			1	0.9			
SSM1 - 8	0.6		0.9	0.6	0.9		1		0.9			
SV - 9	0.8		0.7	0.8	0.7	1						
SpecTestPlat - 10	0.8		1	0.8	1	0.8			0.8			
ExtPlat - 11	0.9		0.7	0.9	0.7	1			1			
ComTestPlat - 12	1		0.9	1	0.9	0.8			0.7			

Figure 2. An Example Portion of *LOCATE's* Priority Weight Summary Table for a Visual Domain

LOCATE now supports 20 such summary tables, including those for link and obstruction functions and for priority weights.

As indicated earlier, these tables do not simply give users an overview of data entered but permit their modification from any of the tables displayed. A user may focus on individual workstations or obstructions, entering data in that format, view all workstations or obstructions for a given domain and enter or modify data in that way, or display all workstations or obstructions for all domains at once and make changes in that context. The key is flexibility and *LOCATE* now provides several modes for data entry that can suit different users with different needs or the same user with different needs at different times.

An optional item was incorporated as part of this work that involved enhancing the speed with which these windows could be opened and made ready for modification. The potential wealth of information in these tables, especially those displaying data for all domains at once, made efficiency a high priority.

Cost Function Summaries. Although these are summary tables as well, they represent results of *LOCATE* analyses rather than data on which those analyses are based, such as described in the previous section.

Early development of the *LOCATE* interface used its output matrices as a basis for generating colour cost displays that provided a much easier way for users to understand where communication costs were occurring in a configuration. The user then could quickly manipulate the configuration to try and help reduce those costs.

Prior to the addition of those displays, users had to examine and compare numerical values, a much more difficult and tedious task, to accomplish the same thing. Of course, *LOCATE* still produced numerical matrices for inspection but they remained cumbersome and unwieldy sources of cost information.

Now, the two representations have been combined in a way that gives the user a more effective context in which to understand simultaneously the sources of communication costs and their mathematical bases. Instead of two separate representations, the numerical values may be displayed within the colour-coded matrices.

Figure 3 shows an example of a colour-coded display much as it was prior to the current work. In that example, high costs appear in red, low costs in white and medium costs in yellow and green. Workstations are shown as receivers of information in the rows and as sources of information in the columns.

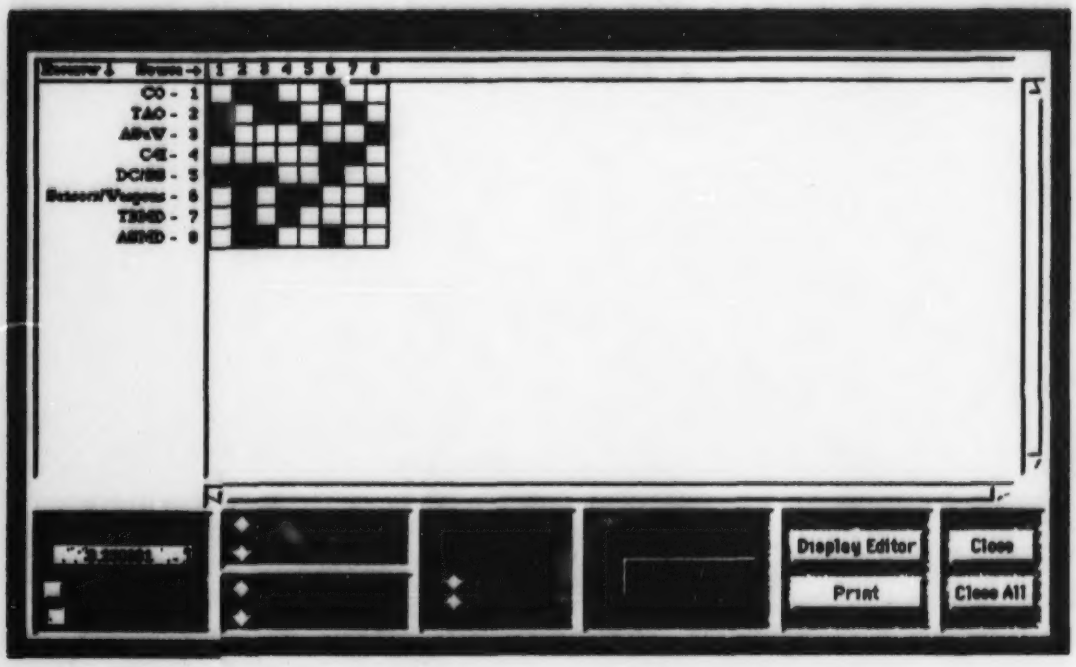


Figure 3. Example Cost Function Output in *LOCATE's* Colour-Coded Cost Display

The example shows cost function results for all domains combined. An overall cost function value is displayed at the bottom left of the window. Also at the bottom left, a toggle "switch" has been added that allows a user to display pairwise cost function values for whatever domain is currently displayed.

Figure 4 illustrates the display of those values for the auditory domain in the same configuration as that used in Figure 3.

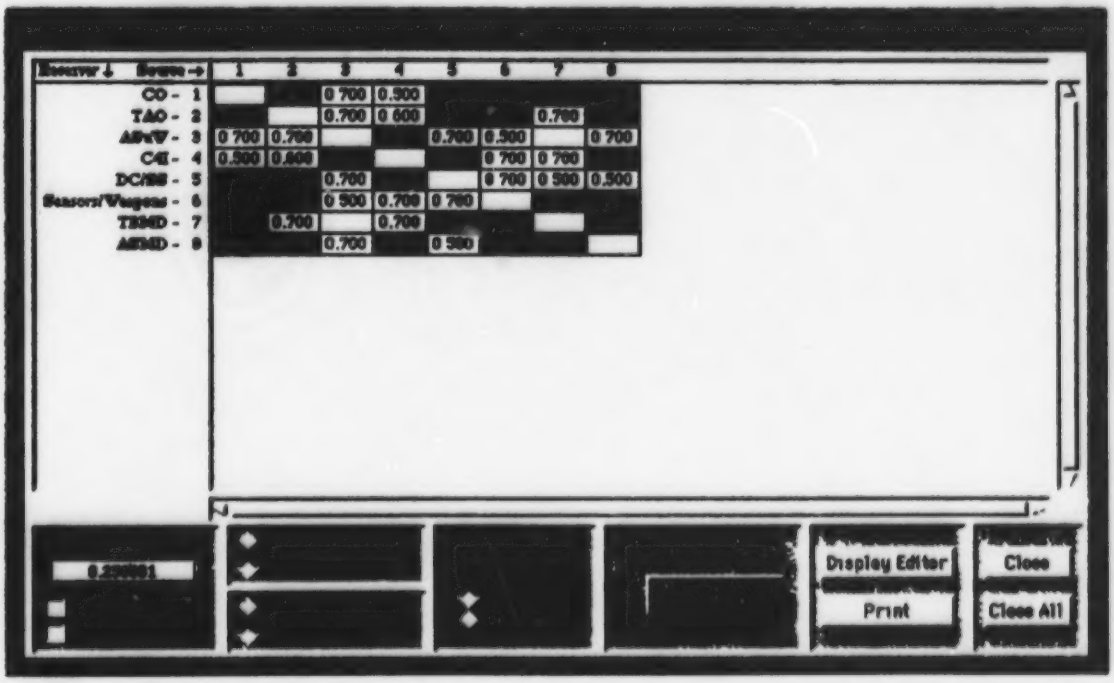


Figure 4. Example Colour-Coded Display of Pairwise Cost Function Values

A user now can examine the numerical values within the context of the colour coding by a simple click of a button. Moreover, the user may display all of the windows for all of the domains of interest, examining and comparing them together and toggling back and forth between numerical values.

Combining the two displays in this manner simultaneously eliminates the need for a separate display of the numerical data and provides a more focused and effective way of examining those data.

Another addition to the display is a facility to reduce clutter by toggling zero values on and off, similar to that used in the priority weight summary tables described above.

Finally, by combining numerical data with colour-cost displays, three of the four optional features identified for the numerical cost function results matrix, listed in the earlier section on "Scope of Work," have come along for free, specifically: 1) the link and weighted costs; 2) the display by domain; and, 3) the deletion of position and rotation information. That last item is preserved, however, in the data exported as part of the attributes and values for workspace, workstations and obstructions. It is described in more detail in the next section.

Import and Export Commands. Close on the heels of the data summary tables and cost function displays in importance is the implementation of an expanded import and export capability for *LOCATE*.

Prior to this work phase, users were able to import and export selected aspects of DXF files. More specifically, they were able to export its key workstation and obstruction objects and import two dimensional representations of AutoCAD and other similar designs saved in standard DXF file format.

In addition, a modification to the import command had allowed for importing priority weight and link function data from other *LOCATE* files, as long as those files contained the same number of workstations as the files into which the data were being imported.

Beyond adding capabilities for DXF files, outlined in an earlier section, *LOCATE* was extended by making the import and export of data more systematic. The best way to understand those additions is to examine the windows *LOCATE* now provides to users for specifying options for importing and exporting data.

Figure 5 shows the import options available. The first option, appearing on the left side of the window, allows a user to specify the file type, which can be either another *LOCATE* file or a comma-delimited text file.

This option permits data to be imported from existing *LOCATE* files and is useful when analysing configurations containing the same workstations. If data are modified in one

configuration, they need not be modified in all others. A user simply modifies the data in one, opens each of the others and imports the data from the modified configuration.

Regarding the comma-delimited file type, users can import data that have been entered in a spreadsheet and saved in a comma-delimited format. They also can create a *LOCATE* file with the workstations to be analysed, export header and default information, and import the resulting file into a spreadsheet for use as a model for data entry. Once complete, the new data can be saved as a comma-delimited file and imported back into *LOCATE*.

The second option in the window in Figure 5 allows the user to identify the type of data to be imported: link function, priority weight, workstation obstruction function or fixed obstruction function data. Only one type of data can be imported at a time but the user can readily import several types, one after the other, by using the Apply button.

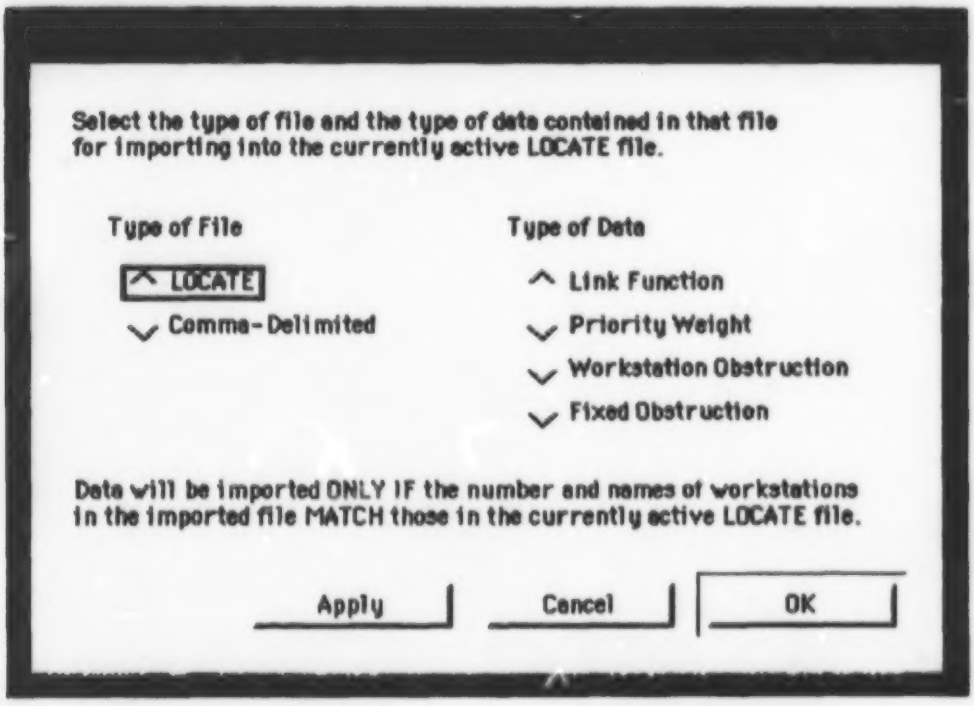

Select the type of file and the type of data contained in that file for importing into the currently active LOCATE file.
Type of File
LOCATE
Comma-Delimited
Type of Data
Link Function
Priority Weight
Workstation Obstruction
Fixed Obstruction
Data will be imported ONLY IF the number and names of workstations in the imported file MATCH those in the currently active LOCATE file.
Apply
Cancel
OK

Figure 5. *LOCATE's* Import Options Window

As the message at the bottom of the window indicates, data will only be imported if the number and names of the workstations and obstructions in the import file match those in the file into which the data are being imported.

The export command invokes the window shown in Figure 6. In addition to exporting the same four types of data discussed for the import command above, users also are able to export attribute and cost function data.

Attribute data includes information about the workspace-as-a-whole including its size, boundaries, zero point, zoom factor, scale and so on. Also exported are the domain weights, which indicate the domains being analysed and their relative importance (weightings), the rules for overlap of objects during optimisation and summary statistics about the number of objects and link types currently in the design.

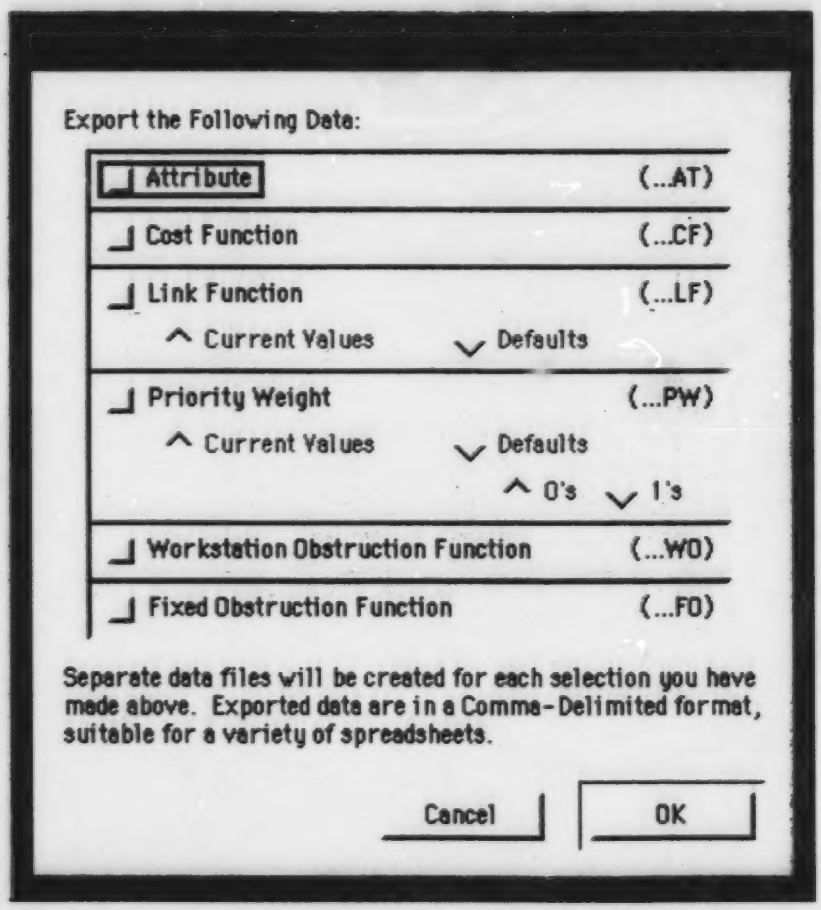


Figure 6. *LOCATE's* Export Options Window

Other attribute data include position and angle of rotation information for workstations, source/receiver nodes within those workstations, and workstation (elemental) and fixed obstructions, minimum and maximum constraints on size of workstations and the radius of their its enclosing boundaries.

The workstation data were originally part of the numerical cost function output, but since those data are now integrated with the colour cost displays, the workstation data have been combined with other attribute data, all of which can be exported and subsequently brought into a spreadsheet for viewing.

The numerical cost function data incorporated as part of the colour-cost displays includes four matrices, which are produced by crossing the normalised and non-normalised options with the link quality and weighted cost options.

Regarding the link function and priority weight data, a user can choose to output current or default values to a comma-delimited file that can then be imported into a spreadsheet for modification. *LOCATE* currently assigns 1's as default priority weights for all pairwise combinations of workstations, considered as both a source and receiver of information, for all domains. Choices in the Export Options Window shown in the above figure, however, providing the flexibility for users to export default values as either 1's or 0's.

Unlike the import command, which requires a user to import different kinds of data one at a time, any combination of data types listed in the window may be exported all at once. The data are saved, however, in separate files under the name of the *LOCATE* file plus a two character ending that identifies the type of file it is. The two character codes appear in the rightmost column of the Export Options Window.

Windows

Modifications to other windows include the Cost Function Checks, Information Update, Links Display, More on *LOCATE*, Obstruction, Print Preview, Tool Library, Source/Receiver (S/R), Workspace, and Workstation Windows as follows:

Cost Function Checks Window. Although some work was done on this window, including the substitution of the Link Function panel with a Workstation panel replacement, a more comprehensive approach to cost function checks needs to be considered in future.

Information Update Window. The conditions under which this window is display have been changed since the window may now be invoked by selecting the "Data Entry" item from the Help Menu.

If the user has selected "Beginner" to describe his knowledge of *LOCATE* in the "About You" Window, the Data Entry Window is displayed whenever he attempts to run a cost function. It is displayed once a session for however long the designation of "Beginner" applies to a user.

Links Display Window. A small problem was corrected allowing for the display of hidden links when any change is made to the active domain weights.

More On LOCATE (MOL)Window. A text size problem was fixed in this window.

Obstruction (Ob) Window. A minor fix was made that involved changing the x-scale and y-scale dimmed attributes in the distance domain to non-editable values for those attributes.

A requirement for copying and pasting obstructions across designs was refined in part as a consequence of AIM's work with the US Navy.

Print Preview Window. A problem was fixed involving multiple refresh while using the Print Preview Screen, which is used to identify the portions of a design a user wishes to have printed. An optional requirement was met that fixed a performance problem occurring when the Print Command was cancelled.

Source/Receiver (S/R) Node Window. Consideration was given to establishing a separate window for Source/Receiver (S/R) Node attributes. Currently, information about S/R nodes is incorporated into the attributes section of the Workstation Window.

After reviewing possible options, it was decided that S/R node information should remain as part of that window and no new menu item was needed in the View Menu to bring up the S/R data.

A useful addition was made, however, to the S/R node information in the Workstation Window where, prior to this, only local coordinates were displayed for S/R nodes. In recent work with US Navy files, it was discovered that global coordinates can be considerable help in positioning workstations. The information is needed because it is the location of the

workstation's S/R node that must remain stable when adjustments are made to items in a workstation or to the workstation boundary.

If an S/R node is not located in the centre of a workstation, it may be necessary to move it to a given location in the workspace using global coordinates. As a consequent of this emergent need, both global and local coordinates now can be displayed in *LOCATE's* Workstation Window. Further, changes made to the position of an S/R node are reflected immediately in the x and y position values of the workstation of which it is a part. Similarly, changes to the position of the workstation are reflected in the x and y position values of its S/R node.

Tool Library. An option of assigning the selection arrow to any portion of the palette was disallowed.

Workspace (WS) Window. Modifying the domains that are to be analysed now closes any open cost display windows.

If the workspace zero point is set to one of the five standard positions and a user changes the dimensions of a workspace, *LOCATE* now automatically sets the new zero point to the same standard position relative to the new dimensions. If the user has customised the zero point, *LOCATE* sets the zero point to the bottom left corner of the new dimensions.

Work on this feature resulted in the addition of a new check, which is performed whenever a user changes the zero point of a workspace. The check is done to make sure that the new zero point falls inside the new dimensions of the workspace.

In the previous version of *LOCATE*, if a user selected one of the standard five positions for the zero point, the button was displayed but, if the user closed and reopened the window, the button selection was no longer displayed. That problem was fixed in this version.

Workstation Window. If a window is opened on a workstation, a different workstation selected in the workspace and its attributes, link functions or priority weights selected for view, the open window is updated to reflect the new selection.

The exception is the priority weights portion of the window, which is updated to reflect itself as both source and (dimmed) receiver, unless first-order priority weights are being analysed.

More Refinements

In addition to work on *LOCATE's* windows, a number of other important features were addressed as part of this work phase.

Demonstration Version and Install Script. These two items were additions to *LOCATE* and were not included in the original requirements for this contract. The chief reason for their addition was to help promote the *LOCATE* tool among potential users.

First, a demonstration version of *LOCATE* was created that set limits on the number of workstations and obstructions that could be created and disabled the "Save," "Save As..." and "Print" commands. In that way, DCIEM is able to provide interested users with a copy of *LOCATE* that includes all of its functionality and interface support but without making it useful for practical applications—an important vehicle for disseminating *LOCATE*.

A demonstration version was created for both the Macintosh and PC platforms and a copy of the latter was sent to a British group working at Farnborough, UK for its presentation to interested parties and for their evaluation.

An install script was also created as a way to help in the dissemination of the *LOCATE* tool. The script simplifies the installation on Mac and PC platforms. *LOCATE* files are automatically organised on a user's hard drive and the appropriate pathways are created so that the application knows the location of relevant files required at startup.

PC Port. In part to promote extended familiarity and use of the *LOCATE* tool, the software was ported to the PC. The current versions of the Macintosh and the PC contain the same functionality.

Alignment. A basic form of alignment has been added to *LOCATE*. Users now are able to select objects and align them according to their top, bottom, left and right edges. They also may centre selected objects arranged horizontally or vertically with respect to one another.

Future work is needed to refine this feature since alignment of certain objects, such as obstructions in more than one workstation, should be disallowed. Other types of alignment, such as equal spacing among aligned objects also should be considered.

Snap-to-Grid. This feature was included to round out some of the basic drawing functionality of the *LOCATE* tool. Users now may opt to have the objects they move snap to an invisible grid or, alternatively, move in smooth increments in the workspace.

Currently, a user does not have control over the size of the grid, which conforms to a standard grid spacing. More specifically, each move in snap-to mode moves the object one-eighth of the distance of a cell of *LOCATE's* grid display.

Scrolling and Nudging. Prior to the current work, objects had to be moved in stages if the user wanted to move them to a position beyond the edge of the design window. That inconvenience has been eliminated and users now may grab objects and drag them to the left, right, top or bottom of the display. Once at the edge of the display, *LOCATE* automatically scrolls the window in the direction in which the user is moving the object, until the mouse button is released.

One aspect of considerable help to any drawing programme has been added, even though not part of the original contract work. That feature is the ability for a user to nudge an object a pixel at a time by first selecting the object and then holding down one of the arrow keys. The procedure allows fine tuning of positions of objects once they have been placed in the general vicinity of the desired location.

System Checks. Although a number of checks had already been incorporated into *LOCATE*, several additional ones were added in this work phase. They include:

- closing all open windows, except the design window, whenever any change is made to active domain weights;
- ensuring that values other than 1's sum to 1.00 for the active domains being analysed;
- disallowing negative domain weights;
- disallowing negative values for certain attributes such as the height and width of objects, the radius of enclosing boundaries, etc;
- disallowing characters other than minus signs and decimal points for numerical entries;
- determining that all priority weight values range between -1.00 and +1.00. These and similar checks also were incorporated into the large summary tables described earlier.

A key question to answer in future is whether all checks other than those for uncommitted priority weights can be done locally as the user is entering data rather than waiting until a cost function is about to be run.

Text and the Text Tool. Recurrent problems with the text tool were addressed and font and font size options now behave appropriately for typing new text and for subsequent modifications to text objects.

A problem that prevented the copying and pasting of text objects was cleaned up and an anomaly produced with repeated selection of a text object resulting in the duplication of part of the edge of the design window inside of the workspace, was fixed.

Preference Settings. After reviewing an option to add a window for preferences, it was decided that much of the information that would go into a Preferences Settings Window was already available in the Workspace Attributes Window. Since establishing a Preferences Window simply would be a matter of transferring the same information to another window, it was decided to leave it where it is for now. In that way the form of the information displayed remains consistent with that of attribute items for other objects, such as those for workstations and obstructions.

As *LOCATE* continues to be refined, a window containing user preferences should be revisited. A number of draw programmes have preferences windows that refer to features that might be useful in *LOCATE*, however, only one or two of those are relevant to the programme's current functionality. For future reference, preferences in packages such as Claris Draw should be considered. A few of those items include:

- rescaling during pasting;
- selection of objects across layers in a multi-layer context;
- user typing, without a selection from the tool palette, that automatically creates a new text object or, alternatively, invokes keyboard shortcuts;
- using smart quotes; showing format codes; applying fractional character widths
- optionally displaying four or eight handles on objects;
- supporting manual or automatic polygon closing;
- supporting automatic smoothing and fill;
- providing customised settings for the degrees to which the Mouse+Shift constraint would conform in object-creation mode;
- allowing for faster/slower gradients in filled objects.

Miscellaneous Items. Intermittent problems with the operation of function graphs were fixed. Although Macintosh menus were to be hidden in Mac versions of *LOCATE*, it was decided that the option to demonstrate *LOCATE's* ability to display a variety of interface looks exceeded a need for eliminating those menus permanently. The PC implementation, of course, does not have those menus.

Future Work

As indicated earlier, several items originally intended for this contract were replaced by others. This was done largely as a consequence of an emergent priority to provide interested and potential users of *LOCATE* with an easy way to explore the tool on their own.

The original requirements also identified a number of "optional" items, reflecting desirable features for *LOCATE*, which were to be addressed if time permitted. Some of those items were in fact incorporated into the current version of *LOCATE* but most remain for future implementation.

The list of items below are those that still need to be considered for incorporation into the *LOCATE* Workspace Layout Design Tool. They include features originally intended for this contract but for which substitutes were made, optional items that time did not permit to be addressed and a few items that emerged during the course of the work or which were previously known but not included in the original requirements.

The first set of items were part of the original requirements but were replaced by other items during the course of the contract. The list also includes some optional items that were part of those requirements. Together, they represent the highest priority items for the development of *LOCATE's* basic functionality.

Alignment

As indicated earlier, basic work on the support of alignment in LOCATE was done during this work phase but some of the more subtle issues were deferred to future work.

- Include possible alignment of objects across container objects, e.g., S/R nodes, and objects within container objects, e.g., Elemental Obstructions (EObs) within a given Workstation or objects within Fixed Obstructions (FObs)

- Other objects should not be able to be aligned across container objects, e. g., EObs across Workstations or generic objects across FObs.

Editing

- Support cutting, deleting & duplicating of multiple objects (some exceptions should apply based on *LOCATE's* manual and automatic grouping rules);
- Support undo for the clearing of multiple objects.

Grouping

- Review the status of S/R nodes with regard to manual and automatic grouping;
- Clean-up problems with manual grouping.

Minimise Boundary

- Explore support for undo for the Minimise Boundary command.
- Fix manual minimisation problem related to rotated objects;
- Explore refinement to Minimise Boundary command around circular objects.

Text and Text Tool

Similar to the alignment feature above, work on the text and text tool solved some problems but deferred others. Included in those yet to be addressed are:

- A thorough testing of text editing to make sure that text objects are behaving as expected;
- Testing of the capabilities for copying and pasting of text objects.

Another useful feature associated with text would be to have some easy way of displaying textual information associated with objects in the workspace, most notably, the name of objects, especially workstations.

Web Element

- Investigate interfaces with Netscape and Microsoft Explorer solution.

Windows

General

- Investigate a need for arrows to navigate to other objects via the object's number in pop-up menus;
- Remove close boxes from windows with OK & Cancel; impose limits on the resizability of windows so that text isn't truncated.
- Incorporate real time display of coordinates;

Cost Function History (CFH) Window

- Allow users to delete all temporary files except those run for current design.
- Number and date and time-stamp each run.

Link Functions (LFs) Window

- Consider consolidation of items in this window to save space;
- When moving between LF windows for different EWs, checks should determine if changes have been made. If so, and if the "Apply" button has not been selected, provide a prompt to the user asking whether to accept or ignore the changes before moving to the new EW.

Workspace (WS) Window

- When the user adds another domain to be analysed and a Cost Display (CD) Window is open, update the pop-up menu for the active domains in the CD window.

In addition to the items in the list above, a number of optional items in the original requirements were not implemented and need to be addressed. Those include:

Automatic Navigation to Newly Selected Objects

- Explore and implement the coordination of selected objects in attribute windows with objects in the design space;

Colour Coding Defaults and Committed Values

- Amber and green colours are to be used to indicate that a value is a default value (amber) vs. a user-entered or user-committed value (green).
  These colours should be used in the summary windows for Link Functions (LFs), Obstruction Functions (Obs) and Priority Weights (PWs). In the Cost Function Checks window, only "uncommitted" default data is shown.

Interface Consistency

- Identify and implement features based on interface rules established for consistency among the DCIEM tools of *LOCATE*, IPME and Safework.

Numerical Cost Function Results Matrix

- Explore optional format for output of data in scientific notation.

Object Creation

- Provide the user with control over whether EObs (and other objects) expand from the top left corner or the centre.

Obstructions

- Design and implement obstructions in a common format that permits them to be moved in and out of EWs, providing for automatic redefinition of an obstruction as either an EOb or FOb depending on where it is located;
- Explore and, if possible, provide an option to model S/R nodes as obstructions;
- Make S/R nodes special items that only can be grouped manually with other objects;
- Review different numbering system for EObs (e.g., 1-1; 1-2; 2-1; etc.).

Palette

- Clean up a problem with Control–Clicking an already selected item in the tool library;
- Make tool icons more consistent with instances placed in *LOCATE's* workspace;
- Design and implement colour 3D icons;
- Explore the use of additional palettes, customised for different problem domains.

Preference Settings

- Explore what current and potential attributes might be incorporated into a "Preferences" file for users, as per the discussion in the earlier section, "Preferences Window."

Renaming Problems (EWs; EObs; FObs)

- Explore refinement of Motif's default behaviour when creating or renaming items in pop-up menus.

Rotation

- Provide a rotation mode to allow for multiple rotations;
- Provide an easily accessible toggle button for that option in the design window.

Text and Text Tool

- Provide support for the creation of text when a user simply starts typing;
- Support a click in the workspace, prior to that typing, that specifies where the text is to be located;
- Allow changing of text by clicking in text box without clicking text tool first;
- Allow resizing of text object using handles;

User Manual

- Extend *LOCATE's* Quick Start Manual to reflect features added.

Windows

Cost Display (CD) Window

- File output and printing of colour cost displays should include design title, domain, cost function, link quality or weighted cost view, and normalised or non-normalised view;
- Explore how a display of percentage values might be computed for the cells in CD windows; if feasible, implement;
- Explore incorporating *LOCATE's* Display Editor directly into its Colour-Cost Display Windows;
- Allow for the selection of a group of cells, using click-hold-drag, and for their associated names and numbers to be highlighted.

Cost Function Checks (CFC) Window

- Extend work on the display of default data;
- Examine and implement other checks in the window that require delayed notification.

Elemental Workstation (EW) Window

- Refine the Priority Weight (1st-Order) portion of the window showing only one panel. Prompt the user that the display is for 1st-order PWs, which are entered for the Distance Domain only.

Links Display (LD) Window

- Correct problems with the display in this window, especially with the display of links with zero values, when all link display options have been turned off.

Summary Windows

- Windows should contain:
  - A column that is either blank or contains a "•." The bullet will indicate that there is a note associated with the decision about the function and weight in that row. In either case, the user should be able to double-click on the cell to display a window that allows the user to enter or edit a note;
- Explore and, if appropriate, incorporate a "Save CF button;
- Add a Commit column having a check box for committing the values in the Link Functions and Obstruction Functions Windows;

Workspace Boundary Tools

- review and, as appropriate, modify corners for walls, circular walls, and, positions in walls for windows and doors.

Miscellaneous Items

- Have undocumented feature for showing/hiding Mac menus in Mac version; default should be that these menus are hidden;
- Remove extraneous display of old Graph Widget Menu items using Control, Command, Option or Alt keys when clicking in the workspace or in windows;
- Add a version number to the Get Info Window in the Macintosh.

It is clear from the above list that a number of other features and refinements need to be added to *LOCATE* to round out its functionality. In spite of the items that still need to be addressed, *LOCATE* is now a useful tool for analysing communication efficiency in a wide variety of contexts. It has most of the basic features of a drawing programme and many special features needed to support effective communication analysis.

In addition to its basic functionality, *LOCATE* is also serving as a testbed for the study of aspects of intelligent aiding in software.

The most important obstacle to effective use of *LOCATE* is the requirement for data entry. Tracking actions, goals and plans of users, as they employ *LOCATE* to help solve

their communication problems, has laid the foundation for helping to overcome this obstacle. Recently, work has been initiated to design and build a collection of Internet agents that can acquire information about Internet sites relevant to the user's data entry problem and interface that information with *LOCATE's* intelligent help components.

A final area of work that needs attention is optimising user-generated workspaces in *LOCATE*. Currently, a simple optimiser has been implemented to provide indications about how a workspace might be altered to increase its communication efficiency. The tool could profit from a closer examination of third party quantitative optimisers and an exploration of more innovative optimisation techniques such as those based on genetic algorithms and other evolutionary approaches.

References

Bass, E.J., Ernst-Fortin, S.T., and Small, R.J. (1997). Knowledge base development tool requirements for an intelligent monitoring aid. *Proceedings of the Tenth Annual Florida Artificial Research Symposium (FLAIRS-97)*, Daytona Beach, FL, May 12-14, pp. 412-416.

Broadbent, G. (1988). *Design in architecture.* London: David Fulton Publishers.

Edwards, J. L. (1990). Intelligent dialogue in air traffic control systems. In J. A. Wise, V. D. Hopkin and M. L. Smith (Eds.), *Automation and systems issues in Air Traffic Control.* New York: Springer-Verlag.

Edwards, J. L., & Hendy, K. (1992). Development and validation of user models in an air traffic control simulation. Paper presented at the Second International Workshop on User Modeling. International Conference and Research Center for Computer Science (IBFI), Dagstuhl Castle, Germany, August 10-13, 1992.

Edwards, J. L., & D. W. Sinclair (2000). Designing intelligence: A case of explicit models and layered protocols. In M. M. Taylor, F. Néel and D. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue II.* North-Holland.

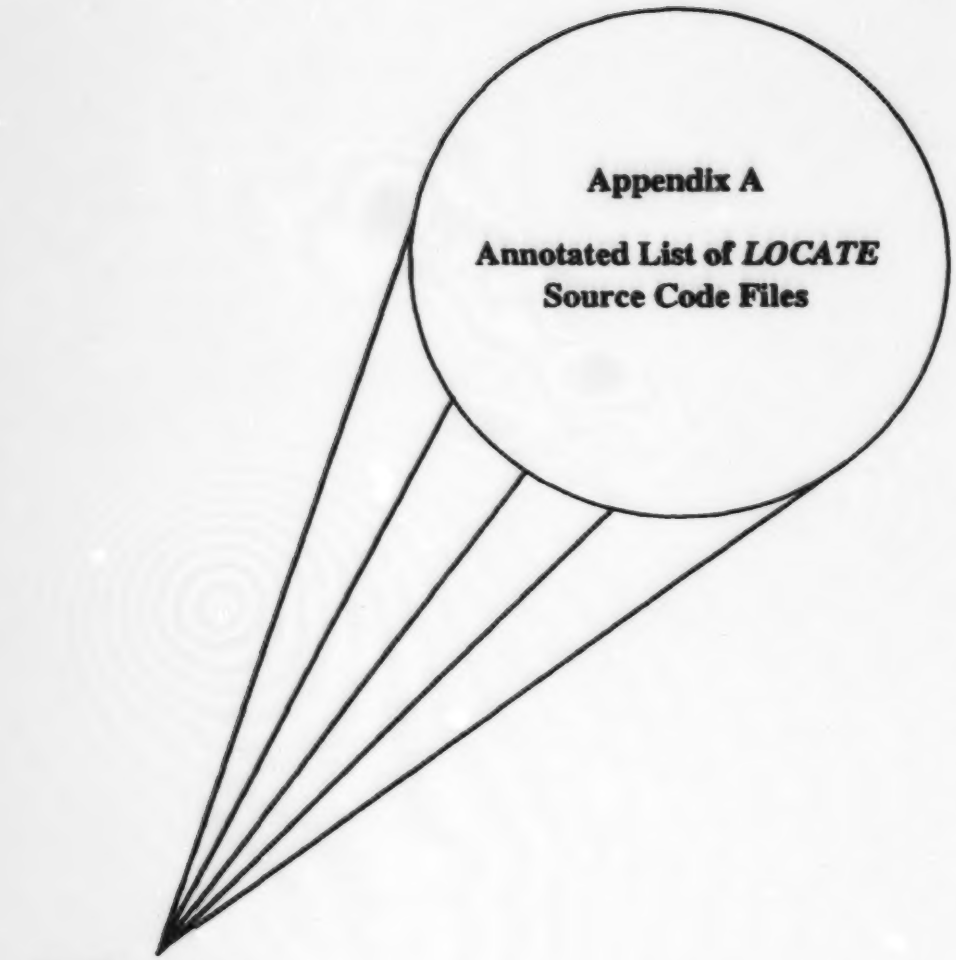
Hendy, K. C. (1984). 'Locate': A program for computer-aided workspace layout. Master's Thesis, Department of Electrical Engineering, Monash University, Clayton, Victoria, Australia.

Hendy, K. C. (1989). A Model for Human-Machine-Human Interaction in Workspace Layout Problems. *Human Factors, 31(5),* 593-610.

Polson, M.C., & Richardson, J.J. (Eds.) (1988). *Foundations of intelligent tutoring systems.* Hillsdale, N.J.: Lawrence Erlbaum.

Spector, J.M., Polson, M.C., and Muraida, D.J. (Eds.) (1993). *Automating instructional design: Concepts and issues.* Englewood Cliffs, N.J.: Educational Technology Publications.

VanLehn, K. (1988). Student Modeling. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 55-78). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.



Appendix A

Annotated List of *LOCATE* Source Code Files

AI→M
Artificial Intelligence
Management and Development Corporation

Notes:

i) .c files are C source code
.h files are C header files
.rc files are Open Interface resource description files

ii) Some extraneous files in the *LOCATE* folders are outdated and will be erased in future.

AEVENT.C

- This file contains the code to handle Apple Events for opening documents by double-clicking in the Finder.

ALIGN.C, ALIGN.RC

- The "Align" module contains the code and Open Interface resources necessary for the "Alignment" window.

ALLOBJINFO.C, ALLOBJINFO.RC

- The "AllObjInfo" module contains the code and Open Interface resources necessary for the "All Objects Info" window.

AOBS.C, AOBS.RC

- The "AObs" module contains the code and Open Interface resources necessary for the "Fixed Obstruction" window.

ASSIGN.C

- Original *LOCATE* C file

CFALERT.C, CFALERT.RC

- The "CFAlert" module contains the code and Open Interface resources necessary for the alert box that appears when there have been changes to the design but no cost function has been run since those changes have been made. The alert box appears before displaying the Cost Function History window .

CFBROWSE.C, CFBROWSE.RC

- The "CFBrowse" module contains the code and Open Interface resources necessary for the "Cost Function History" window.

CFCHECK.C, CFCHECK.RC

- The "CFCheck" module contains the code and Open Interface resources necessary for the "Cost Function Checks" window.

COSTCOLR.C, COSTCOLR.H, COSTCOLR.RC

- The "CostColr" module contains the code and Open Interface resources necessary for the "Cost Display Editor" window.

COSTDISP.C, COSTDISP.H, COSTDISP.RC

- The "CostDisp" module contains the code and Open Interface resources necessary for the "Cost Display" window.

COSTFN.C, COSTFN.RC

- The "CostFn" module contains the code and Open Interface resources necessary for the "Cost Function" window.

DRAWROTD.C, DRAWROTD.H

- C code for handling the drawing of rotated objects

DXF.H

- Header file with DXF format constants

DXFOPT.C, DXFOPT.H

- The "DXFOpt" module contains the code and Open Interface resources necessary for the "DXF Import Options" window.

EDITOR.C, EDITOR2.C, EDITOR.RC

- The "Editor" module contains the code and Open Interface resources necessary for the main *LOCATE* window (includes code for Diagrammer, palette, rulers).

EVAL1.C

- Original *LOCATE* C file

EWATTR.C, EWATTR.H, EWATTR.RC

- The "EWAttr" module contains the code and Open Interface resources necessary for the "Workstation" window, which now includes separate tabbed sections for Attributes, Link Functions and Priority Weights.

EXPORT.C, EXPORT.RC

- The "Export" module contains the code and Open Interface resources necessary for the "Export Data" window.

EXTERN.H

- Original *LOCATE* header file

FOBSUMM.C, FOBSUMM.RC

- The "FObSumm" module contains the code and Open Interface resources necessary for the "Fixed Obstruction Summary" window.

FORMAT.H

- Original *LOCATE* header file

FUNCT1.C

- Original *LOCATE* C file

GOALOBJ.CPP
- Contains code for defining and handling the C++ goal object

HEADER.DXF
- Contains information that gets added to all exported DXF files

HELPALRT.C, HELPALRT.RC
- The "HelpAlrt" module contains the code and Open Interface resources necessary for displaying the "Help Message" windows.

HELPMORE.C, HELPMORE.RC
- The "HelpMore" module contains the code and Open Interface resources necessary for displaying the "Help Message" windows containing the "More" button.

IMPORT.C, IMPORT.RC
- The "Import" module contains the code and Open Interface resources necessary for the "Import Data" window.

IMPRTDXF.C
- C code for handling the importing of a workspace from DXF format

INFOUPD.C, INFOUPD.H, INFOUPD.RC
- The "InfoUpd" module contains the code and Open Interface resources necessary for the "Information Update" window.

INFOWIN.C
- C code for handling the "Object Info" window

LFSUMM.C, LFSUMM.RC
- The "LFSumm" module contains the code and Open Interface resources necessary for the "Link Function Summary" window.

LINKDISP.C, LINKDISP.RC
- The "LinkDisp" module contains the code and Open Interface resources necessary for the "Link Display" window.

LOCATE.C
- Based on the original *LOCATE*.C file, this contains the code necessary for loading in a workspace and for computing the cost function.

LOCATE.DAT
- Open Interface compiled resources that are used by *LOCATE* at run-time.

LOCATE
- The *LOCATE* application

LOCATE.H
- Original *LOCATE* header file

LOCATE.μ
- *LOCATE* project for CodeWarrior 11

LOCATE.RC
- Open Interface resources in text format

LOCNEW.C
- C code for handling the creation of a new workspace

LOCNEWEW.C
- C code for handling the creation and deletion of workstations and obstructions

LOCSAVE.C
- C code for handling the saving of a workspace

LOCSAVEDXF.C
- C code for handling the saving of a workspace in DXF format

MAIN.C, MAIN.RC
- The "Main" module contains the "main" function which starts up the application.

MISC.C
- Original *LOCATE* C file

MISCRSRC.RC
- The "MiscRsrc" module contains Open Interface resources needed by the application (primarily menu and icon resources).

MULTIOBJ.C, MULTIOBJ.RC
- The "MultiObj" module contains the code and Open Interface resources necessary for the "Multiple Object Creation" window.

NEWUSER.C, NEWUSER.H, NEWUSER.RC
- The "NewUser" module contains the code and Open Interface resources necessary for the "About You" window.

OBSUMM.C, OBSUMM.RC
- The "ObSumm" module contains the code and Open Interface resources necessary for the "Elemental Obstruction Summary" window.

OPT.C
- Optimizer code for changing positions and angles

OPTIM.C
- Original *LOCATE* C file

OPTOPT.C, OPTOPT.RC
- The "OptOpt" module contains the code and Open Interface resources necessary for the "Optimizer Options" window.

OPTSET.C, OPTSET.RC
- The "OptSet" module contains the code and Open Interface resources necessary for the "Optimizer Settings" window.

OPTSTAT.C, OPTSTAT.RC
- The "OptStat" module contains the code and Open Interface resources necessary for the "Optimizer Status" window.

OPTSWAP.C
- Optimizer code for swapping workstations

ORIGIN.C
- Original *LOCATE* C file

OTHEROBJ.C, OTHEROBJ.RC
- The "OtherObj" module contains the code and Open Interface resources necessary for the "Other Object" window.

OUTPUT.C
- Original *LOCATE* C file

PALEDIT.C, PALEDIT.RC
- The "PalEdit" module contains the code and Open Interface resources necessary for the "Palette Editor" window.

PLANS.C, PLANS.RC
- The "Plans" module contains the code and Open Interface resources necessary for the "Plans" window.

PLANREC.C
- The "PlanRec" module contains the code that creates the plan fragments and performs plan recognition.

PRINTPREV.C, PRINTPREV.RC
- The "PrintPrev" module contains the code and Open Interface resources necessary for the "Print Preview" window.

PWSUMM.C, PWSUMM.RC

- The "PWSumm" module contains the code and Open Interface resources necessary for the "Priority Weights Summary" window.

RULEEW.C, RULEEW.H, RULEEW.RC

- The "RuleEW" module contains the code and Open Interface resources necessary for the window that informs the user about double-clicking to bring up Workstation attributes.

RULEGEN.C, RULEGEN.H, RULEGEN.RC

- The "RuleGen" module contains the code and Open Interface resources necessary for the window that informs the user about double-clicking to bring up object attributes.

RULEOB.C, RULEOB.H, RULEOB.RC

- The "RuleOB" module contains the code and Open Interface resources necessary for the window that informs the user about double-clicking to bring up Obstruction attributes.

RULEOO.C, RULEOO.H, RULEOO.RC

- The "RuleOO" module contains the code and Open Interface resources necessary for the window that informs the user about double-clicking to bring up Other Object attributes.

RULER.C, RULER.RC

- The "Ruler" module contains the code and Open Interface resources necessary for the "Ruler" window.

SMRTHELP.C, SMRTHELP.RC

- The "SmrtHelp" module contains the code and Open Interface resources necessary for the "Smart Help" window.

SPLASH.C, SPLASH.RC

- The "Splash" module contains the code and Open Interface resources necessary for the startup screen.

SPLASH2.C, SPLASH2.H, SPLASH2.RC

- The "Splash2" module contains the code and Open Interface resources necessary for the "More on *LOCATE*" window.

START.C, START.RC

- The "Start" module contains the code and Open Interface resources necessary for the usability "Start" window.

STARTUP.C, STARTUP.RC

- The "Startup" module contains the code and Open Interface resources necessary for the help reminder at startup.

STARTLOG.C, STARTLOG.RC

- The "StartLog" module contains the code and Open Interface resources necessary for the startup window that allows the user to enter a user name for the help system.

SYSMODL.C, SYSMODL.RC

- The "SysModl" module contains the code and Open Interface resources necessary for the "System Model" window.

TASKMODL.C, TASKMODL.RC

- The "TaskModl" module contains the code and Open Interface resources necessary for the "Task Model" window.

TREE.C

- The "Tree" module contains the code that constructs and searches the plan hierarchy.

USERMODL.C, USERMODL.RC

- The "UserModl" module contains the code and Open Interface resources necessary for the "User Model" window.

WEBBROWS.C, WEBBROWS.H, WEBBROWS.RC

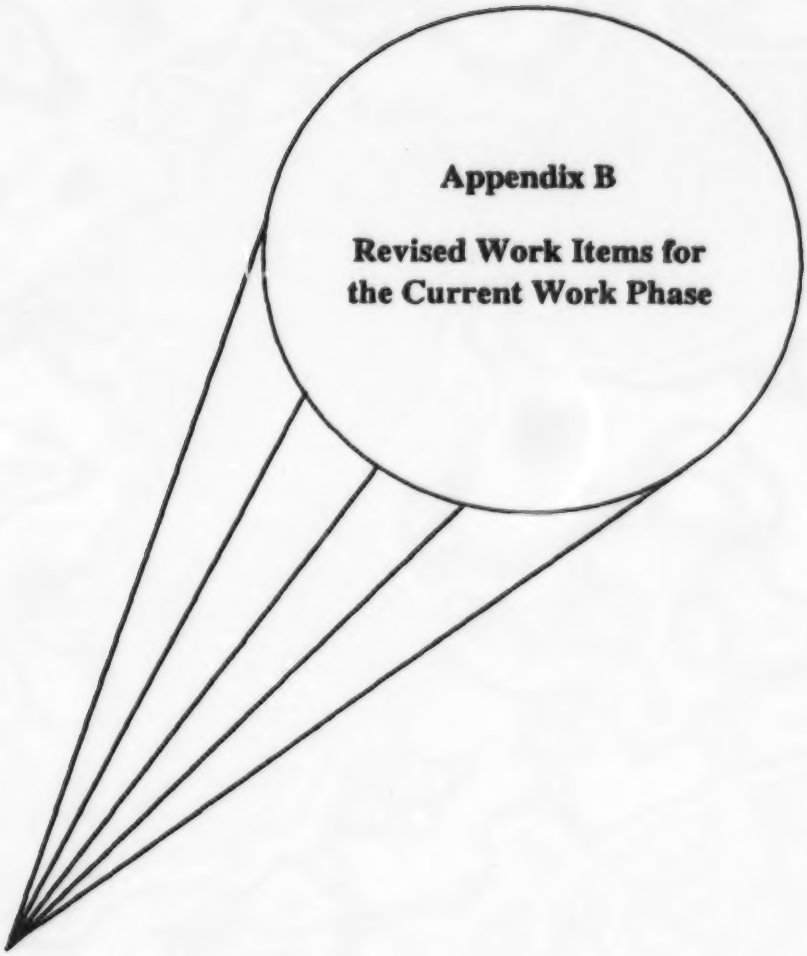
- The "WebBrows" module contains the code and Open Interface resources necessary for the "Web Browser" window.

WOBS.C, WOBS.RC

- The "WObs" module contains the code and Open Interface resources necessary for the "Elemental Obstruction" window.

WSATTR.C, WSATTR.RC

- The "WSAttr" module contains the code and Open Interface resources necessary for the "Workspace Attributes" window.



Appendix B

Revised Work Items for the Current Work Phase

Artificial Intelligence
Management and Development Corporation

I. WORK ITEMS NOT PART OF ORIGINAL REQUIREMENTS

Demo Versions of LOCATE

- Build a demo version of *LOCATE* for the PC and for the Mac.

Install script

- Define requirements for creating an install script for *LOCATE* for Mac and PCs;
- Implement for the PC first, then the Mac.

PC Port

- Port *LOCATE* to the PC with all features implemented under the current contract.

II. RETAINED WORK ITEMS FROM ORIGINAL REQUIREMENTS

CAD Extensions

- Recent work with imported DXF files has revealed a problem with the scaling of some of *LOCATE's* generic objects, which needs adjusting, especially when the size of the imported file is much larger or much smaller than the user-specified *LOCATE* workspace file;
- Other needed extensions include support for exporting and importing of those generic objects.

Import and Export PW data

- Provide a facility for importing and exporting Priority Weight Data in a Spreadsheet form. The user will create a WS in *LOCATE* with the workstations for which he will enter or has entered data. The user should give workstations names. He will then use a new *LOCATE* command to export a spreadsheet file with header information that identifies the workstation names and all of the priority weight data. If the user has already input PW data, those data will be exported in the file. If he has only default data or wishes to revert to default data, he will be given the option to do so. In addition, he can specify whether he wants all of the default data to be 0's or 1's.

Other Interface Refinements

Alignment

- Provide support for the alignment of objects in *LOCATE's* workspace.
  Alignment should not be permitted for objects contained within other objects, e.g., Elemental Obstructions (EObs) contained in two or more Elemental Workstations (EWs) should not be able to be aligned;
  [Note: The above item is one of two that were partially replaced with " Import and Export PW data", above.]

Grid (Snap-to)

- Add a snap-to grid capability.

Importing Link Function, Obstruction Function and Priority Weight data

- Support the transferring of Function and Priority Weight data among designs.
  - Allow users to import Link Function, Priority Weight and Obstruction Function data from other designs; both designs match in terms of the number and names of their EWs;
  - In the case where there is more than one EW, EOb or FOb, all must have unique names.

Numerical Cost Function Results Matrix

- Add names and numbers of EWs for rows and numbers for columns;
- Produce a "Total" matrix to add to the output of the existing numerical matrices.

Preference Settings

- Create a "Preferences" file with selected setting options.

Printing

- Fix multiple refresh problem.

Scrolling

- Dragging objects off the screen in any direction should cause automatic scrolling in that direction.

System Checks

- Update (or close) all open windows when changes are made to the active domain weights;
- Check for values of domain weights other than 1's;
- Check for negative domain weights;
- Check for negative values for numbers;
- Check for commas instead of periods in numbers.

Text and Text Tool

- Fix problems associated with current text editing;
- Fix problems with current copy and paste of text objects.
  [Note: The above item is one of two partially replaced with " Import and Export PW data", above.]

Tool Library

- Delete the choice of the "Selection Tool" from the tool library.

Windows

Cost Display (CD) Window

- Output of colour cost displays should include design title, domain, cost function, link quality or weighted cost view, and normalised or non-normalised view;
- Add an "All" to display pop-up inside each colour CD window, supporting similar functionality to the "Close All" button.

Cost Function Checks (CFC) Window

- Extend earlier work on the display of default data in the Cost Function Checks Window.

Elemental Workstation (EW) Window

- Update already open EW Windows when another EW is double-clicked in the workspace; (E)

Information Update (IU) Window

- Determine rules to govern the display of the IU window.
  Consider the following conditions under which that window might be shown:
  1. When Smart Help is active.
  2. Once only, prior to each successful attempt to run a CF.

Links Display (LD) Window

- Correct minor problems with the display in this window.

More On *LOCATE* (MOL) Window

- Correct a minor display problem in this window.

Obstruction (Ob) Windows

- Make the x-scale and y-scale values in the Obstruction windows non-editable.

S/R Node (S/R) Window

- Add windows for S/R nodes, if appropriate.
  [Note: This item was replaced with displays of both local and global x,y and angle attributes of S/R nodes. The nodes]

Summary Windows

- Create summary windows for 1) Link Functions; 2) Obstruction Functions; and, 3) Priority Weights;
- Make the windows scrolling and the values in the windows editable;
- Windows should contain pop-up menus for link function selection;
- Incorporate checks that PWs range from -1.00 to +1.00;
- Explore incorporating "Cost Function Value Display" ~~and "Save CF button;~~

Workspace (WS) Window

The original of this item is eliminated and the following substituted:

- When the user adds another domain to be analysed and CD Windows are open, close them.

Zero Point of Workspace

- By default, the zero point is set to the centre of the WS. When changing workspace dimensions, automatically update the selected zero point;
- Highlight the radio button defining the workspace zero point;
- Fix problems associated when revising a workspace's dimensions to create a smaller workspace after objects have been created that will fall outside the new workspace.

Miscellaneous Items

- Identify and fix intermittent problems with function graphs;
- Hide Mac menus in Mac version.

III. WORK ITEMS DELETED FROM ORIGINAL REQUIREMENTS

Other Interface Refinements

Editing

- Support cutting, deleting & duplicating of multiple objects (some exceptions apply based on *LOCATE's* manual and automatic grouping rules.);
- Support undo for the clearing of multiple objects.

Grouping

- Review the status of S/R nodes with regard to manual and automatic grouping; (E)
- Clean-up problems with manual grouping. (O)

Minimise Boundary

- Explore support for undo for the Minimise Boundary command.
- Fix manual minimisation problem related to rotated objects; (O)
- Explore refinement to Minimise Boundary command around circular objects. (O)

Web Element

- Investigate interfaces with Netscape and Microsoft Explorer solution.

Windows

General

- Investigate the need for arrows to navigate to other objects via the object's number in pop-up menus; if needed, implement;
- As needed, remove close boxes from windows with OK & Cancel; impose limits on the resizability of windows so that text isn't truncated.
- Incorporate real time display of coordinates; (O)

Cost Function History (CFH) Window

- Allow users to delete all temporary files except those run for current design.
- Number and date and time-stamp each run. (O)

Link Functions (LFs) Window

- Consider consolidation of items in this window to save space;
- When moving between LF windows for different EWs, checks should determine if changes have been made. If so, and if the "Apply" button has not been selected, provide a prompt to the user asking whether to accept or ignore the changes before moving to the new EW.

Workspace (WS) Window (see substitution above)

- When the user adds another domain to be analysed and a CD Window is open, update the pop-up menu for the active domains in the CD window. (E)

DOCUMENT CONTROL DATA SHEET

1a. PERFORMING AGENCY	2. SECURITY CLASSIFICATION
Artificial Intelligence Management and Development Corporation, 206 Keewatin Avenue, Toronto, ON M4P 1Z8, CANADA	UNCLASSIFIED -
1b. PUBLISHING AGENCY DCIEM	

3. TITLE

(U) CAD extensions and other refinements to the LOCATE Workspace Layout Tool - Final report

4. AUTHORS

Edwards, Jack L.

5. DATE OF PUBLICATION	6. NO. OF PAGES
May 1 , 2000	49

7. DESCRIPTIVE NOTES

8. SPONSORING/MONITORING/CONTRACTING/TASKING AGENCY
Sponsoring Agency:
Monitoring Agency:
Contracting Agency : DCIEM
Tasking Agency:

9. ORIGINATORS DOCUMENT NUMBER	10. CONTRACT GRANT AND/OR PROJECT NO.	11. OTHER DOCUMENT NOS.
Contract Report CR 2001-018	W7711-9-7600-001-TOR; 6kc12	AC207

12. DOCUMENT RELEASABILITY

Unlimited distribution

13. DOCUMENT ANNOUNCEMENT

14. ABSTRACT

(U) The LOCATE workspace layout tool is a mature tool useful for practical design applications. It has been used to evaluate bridge configurations for a Tribal class (DDH280) destroyer of the Canadian Forces, to analyse proposed ship designs for the US Navy as part of their Surface Combatant-21 (SC-21) project and is currently being considered as part of work at the Canadian Regional Operations Centre. Other potential application areas include office and manufacturing plant layouts, command posts, aircraft cockpits and even computer screens. The work for this contract comprised refinements to the LOCATE workspace layout tool. A key concern was extending LOCATEs ability to deal with files created in other CAD packages and with creating similar output files for use in those same packages. A second important goal was to expand and refine select features of LOCATE. One feature of particular interest was the extension of overview summaries of link and obstruction functions and priority weights so that users will be able to examine, compare and edit those data in a convenient form. Finally, specifications for C + + data structures, such as those for goals, plans and models, were used in a continuing effort to build the kind of infrastructure needed to support a truly intelligent LOCATE.

15. KEYWORDS, DESCRIPTORS or IDENTIFIERS

(U) HUMAN ENGINEERING TOOLS; HUMAN MODELLING; WORKSPACE LAYOUT; WORKSPACE DESIGN; FACILITY LAYOUT; COMPUTER-AIDED DESIGN; INTELLIGENT HELP; ADAPTIVE INTERFACES; LOCATE; EXPLICIT USER MODELS